UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/675,783	09/30/2003	Frank Eliot Levine	AUS920030486US1	6547

35525 7590 04/17/2008

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
MITCHELL, JASON D	
ART UNIT	**PAPER NUMBER**
2193	
NOTIFICATION DATE	**DELIVERY MODE**
04/17/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptonotifs@yeeiplaw.com

Office Action Summary	**Application No.** 10/675,783	**Applicant(s)** LEVINE ET AL.	
	Examiner Jason Mitchell	**Art Unit** 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 February 2008*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *6-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *6-19* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

This action is in response to a Request for Continued Examination filed 2/27/08.

Claims 6-19 are pending in this application.

Response to Arguments

Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 6-19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 6 recites the limitation "each functional unit of said processor" in line 17. There is insufficient antecedent basis for this limitation in the claim. For the purposes of this examination "each functional unit of said processor" is interpreted to refer to the instruction cache and sequencer unit of the processor (see lines 6-7).

Claims 11 and 19 make similar recitations and are also rejected for the reasons discussed above.

Claims 7-10 and 12-18 depend from claims 6 and 11 and are also rejected for the reasons discussed above.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 6-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,966,057 to Lueh (Lueh) in view of US 2003/0225917 to Partamian et al. (Partamian) in view of US 2004/0030870 to Buser (Buser) in view of US 5,751,942 to Christensen et al. (Christensen) in view of official notice.

Regarding Claims 6 and 11: Lueh discloses profiling an application in a data processing system, the method comprising:

each one of a plurality of individual instructions associated with an indicator that indicates that each one of the plurality of instructions needs to be monitored (col. 6, lines 1-3 “a location map where the original code needs to be replaced with a branch or trap instruction”);

an instruction cache (col. 8, lines 43-47 "instruction cache”) and instructions for using said indicator to detect execution of each one of the plurality of instructions, wherein execution of instructions, which are not associated with the indicator, is not detected (col. 6, lines 1-3 “original code needs to be replaced with a branch or trap instruction”; col. 5, lines 57-67 “Breakpoints can be implemented using … trap patching and code patching” note that instructions not indicated in the “location map” will not be monitored and thus their execution will not be detected);

the instruction cache sending a signal to a performance monitor unit responsive to the instruction cache detecting said indicator in a particular one of the plurality of instructions (col. 7, lines 9-16 “instrumentation code passes necessary information to a run-time library function”; col. 6, lines 1-3 “a branch or trap instruction”), wherein the particular one of the plurality of instructions is located in a routine (col. 3, lines 26-31 “A code segment may represent … a routine”), and further wherein the signal is not sent to the performance monitor unit for instructions that are not associated with the indicator (col. 6, lines 1-3 “original code needs to be replaced with a branch or trap instruction”; col. 5, lines 57-67 “Breakpoints can be implemented using … trap patching and code patching” note that instructions not indicated in the “location map” will not be monitored and thus their execution will not be detected), and further wherein the signal indicates that the particular one of the plurality of instructions is being executed (col. 5, lines 57-67 “stop program execution at desired locations"), and still further wherein the performance monitor unit is coupled to each functional unit of said processor (col. 3, lines 31-34 “A code segment may be coupled to another code segment or a hardware circuit by passing and/or receiving information");

the performance monitor unit counting events that are associated with an execution of only said plurality of instruction that are associated with the indicator responsive to the performance monitor unit receiving the signal (col. 4, lines 60-65 “methods that are identified as hot methods”; note that execution of instructions not indicated in the “location map” will not be monitored, i.e. no breakpoint will be encountered);

collecting means for collecting data from the performance monitor unit (col. 4, lines 60-65 “The profile data representation 340 includes statistics ... collected by a counter 345");

using means to identify a caller of the routine (col. 5, lines 10-15 "a mechanism to identify and access the caller's frame context");

the instruction cache for determining whether the particular one of the plurality of instructions is ‘hot’ (col. 4, lines 60-65 “methods that are identified as hot methods based on the collected profiling information”); and

the instruction cache, responsive to the particular one of the plurality of instructions having been identified as ‘hot’, generating an interrupt to pass control to a monitoring program (col. 9, lines 14-35 “The execution 640 includes ... executing an event hook function for an event corresponding to the +field watch”; col. 7, lines 5-8 “To support the watch points for fields, the JIT compiler interrupts the execution ... to call the event hook function”), wherein the monitoring program identifies information regarding a caller of the routine (col. 5, lines 11-16 “the JIT compiler provides a mechanism to identify and access the caller’s frame context, referred to as unwinding stack frame.”).

Lueh does not explicitly disclose determining whether the instruction is ‘hot’ comprises determining whether the instruction has been executed more often than a threshold value.

Partamian teaches a method of determining whether an instruction is ‘hot’ comprising determining whether the instruction has been executed more often than a threshold value (par. [0018] “JVM 1120 includes a threshold to determine whether a method is hot or not.”)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to compare the profiling information collected by Lueh’s “counter 345” (see, col. 4, lines 60-65) to a threshold value, as taught by Partamian (par. [0018]) as an obvious and commonly used method to identify hot methods for recompilation (Lueh col. 4, lines 60-65 “re-compiles methods that are identified as hot methods”).

The Lueh-Partamian combination does not disclose the indicator stored in at least one existing spare bit in each one of the plurality of individual instructions. Instead the Lueh reference discloses implementing breakpointing using one of two methods (i.e. col. 5, lines 57-67 “trap patching and code patching”).

Further Lueh discloses an instruction cache (col.8, lines 43-47 "instruction cache") and that other aspects of his system "may be implemented by hardware" (see col. 3, lines 23-24) and which may be coupled to ... a hardware circuit by passing and/or receiving information" (col. 3, lines 31-38). Partamian teaches a cpu which contains a plurality of processors (CPU 304 may include … a plurality of processors"). However the Lueh-Partamian combination does not explicitly disclose the instruction cache outputs to a sequencer unit that subsequently outputs to execution units all three of which are included in a processor.

Buser teaches a third method of breakpointing in which the indicator is stored in at least one existing spare bit in each one of the plurality of individual instructions (par. [0004] providing an instruction field in every instruction … actions are performed … based on the value of the halt identifier field").

Further, Buser teaches an instruction cache (Fig. 1, 1001; par. [0018] "Instruction memory 101 is comprised of instructions") coupled to a processor (Fig. 1, "CPU 0" 1000), that outputs said plurality of instructions to a sequencer unit (Fig. 1, "Prefetch and Decode Logic" 1006) that outputs the plurality of instructions to an execution unit (Fig. 7, Execution Unit 1007) wherein the sequencer unit and execution units are included in the processor (Fig. 1, "CPU 0" 1000).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Buser's third method of breakpointing in the Lueh-Partamian combination (e.g. Lueh col. 5, lines 57-67) as an alternate means of providing the desired functionality (i.e. breakpoints). One of ordinary skill in the art would have been able to implement the modified system with predictable results.

It would additionally have been obvious to one of ordinary skill in the art at the time the invention was made to implement the suggested profiling and optimization functionality (e.g. Lueh col. 4, lines 60-65 "methods that are identified as hot methods based on the collected profiling information and generates the optimized native code 360") with an instruction cache (Leuh col. .8, lines 43-47 "instruction cache"; Buser Fig. 1, 1001) that outputs said plurality of instructions to a sequencer unit (Buser Fig. 1, “Prefetch and Decode Logic” 1006) that outputs the plurality of instructions to execution units (Buser Fig. 7, Execution Unit 1007; CPU 304 may include ... a plurality of processors"), wherein the sequencer unit and the execution units are included in a processor (Fig. 1, “CPU 0” 1000). Those of ordinary skill in the art would have been motivated to implement the system in such a way to achieve the various performance gains normally associated with a hardware implementation over a software implementation.

The Lueh-Partamian-Buser does not explicitly disclose the instruction cache (e.g. Lueh col.8, lines 43-47 "instruction cache"; Buser Fig. 1, 1001) is included in the processor.

Christensen teaches an instruction cache (Fig. 1, ICache 120) included in a processor (Fig. 1, “Target Microprocessor” 100).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the instruction cache (e.g. Lueh col.8, lines 43-47 "instruction cache”; Buser Fig. 1, 1001) in the processor (e.g. Christensen Fig. 1 “Target Microprocessor” 100; Buser Fig. 1, “CPU 0” 1000). Those of ordinary skill in the art would have been motivated to do so in order to take advantage of the superior access time of cache memory. Specifically, such a configuration would avoid the need to retrieve instructions from memory (Christensen col. 3, lines 60-65 “instructions enter processor 100 through instruction bus 30 and are stored in instruction cache 120 until they are required by core 110).

Lueh discloses using means to identify a caller of the routine (col. 5, lines 10-15 "a mechanism to identify and access the caller's frame context"); however, the Lueh-Partamian-Buser-Christensen combination does not disclose the using means uses the collected profile data to identify a caller of the routine.

Official notice is taken that identifying a caller of a routine is a common use of profiling data.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Lueh's collected profile data (col. 4, lines 60-65 "The profile data representation 340 includes statistics ... collected by a counter 345") to determine a caller of a routine. Those of ordinary skill in the art would have been motivated to do so in order to provide the data (e.g. in the form of a call graph) to a user for analysis.

Regarding Claims 7 and 12: The rejections of claims 6 and 11 are incorporated, respectively; further Lueh discloses:

examining a call stack upon generation of the interrupt (col. 5, lines 11-16 "unwinding stack frame."); and

identifying a caller of the routine from an examination of the call stack (col. 5, lines 11-16 "the JIT compiler provides a mechanism to identify and access the caller's frame context").

Regarding Claims 8 and 13: The rejections of claims 6 and 11 are incorporated, respectively; further Lueh discloses the information includes at least one of a caller of the routine (col. 5, lines 11-16 "the JIT compiler provides a mechanism to identify and access the caller's frame context") and a number of instructions executed in the routine.

Regarding Claims 9 and 14: The rejections of claims 6 and 11 are incorporated, respectively; further Lueh discloses:

generating a call graph from the information (col. 5, lines 11-16 "unwinding stack frame.").

Regarding Claims 10 and 15: The rejections of claims 6 and 11 are incorporated, respectively; further Lueh discloses:

selecting the caller of the routine for analysis based on the information gathered by the monitoring program (col. 5, lines 14-15 "The stack unwinding process starts with a frame context of the caller").

Regarding Claims 16 and 18: The rejection of claims 6 and 11 are incorporated; further the Lueh-Partamian combination discloses:

indicating that each execution of each one of the plurality of instructions should be counted (Lueh col. 6, lines 1-3 "a location map where the original code needs to be replaced with a branch or trap instruction");

identifying a threshold value (Lueh col. 4, lines 60-65 "methods that are identified as hot methods based on the collected profiling information"; Partamian par. [0018] "JVM 1120 includes a threshold to determine whether a method is hot or not."); and

a counter to count a number of times each one of the plurality of instructions is executed (Lueh Fig. 3 counter 345; col. 4, lines 60-65).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the indication that executions of an instruction should be counted (Lueh col. 6, lines 1-3), threshold value (Partamian par. [0018]), and a counter (Lueh col. 4, lines 60-65) in a first second and third one of a plurality of existing spare bits in each one of the plurality of instructions (Buser par. [0004] providing an instruction field in every instruction … actions are performed … based on the value of the halt identifier field" see the rejection of claims 6 and 11). One of ordinary skill in the art would have been able to implement the modified system with predictable results.

Regarding Claim 17: The rejection of claim 16 is incorporated; further Buser discloses the use of registers for controlling a meaning of each one of the plurality of bits (par. [0021] "the value of CPU halt identifier field 1002 for that instruction is compared to the value of … a special register within the CPU").

Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,966,057 to Lueh (Lueh) in view of US 5,896,538 to Blandy et al. (Blandy) in view of US 2004/0030870 to Buser (Buser) in view of US 5,751,942 to Christensen et al. (Christensen) in view of official notice.

Regarding Claim 19: Claim 19 primarily recites a combination of the limitations addressed separately in claims 6-10 and 16-17.

Blandy teaches a threshold value used to identify a hot method (col. 3, lines identifies the hot modules … After a module has been called a certain number of times") similar to, and resulting in the same obvious modifications as the Partamian teaching relied upon in the rejection of claims 6-10 and 16-17.

Further, Blandy teaches a “performance monitor may track the cycle time for a module” (col. 3, lines 8-10). Accordingly it would have been obvious to one of ordinary skill in the art at the time the invention was made to use cycle time for the threshold value as an alternative means of determining a hot method.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason Mitchell/
Jason Mitchell

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

3/17/08